۸۱۰۱۰۰۰۸۴

محمد امانلو

تکلیف دستی سوم درس طراحی الگوریتم

استاد اسدپور

فصل ۵ . سوال ۵ . صفحه ۱۷۴ . تمرین ۱ فصل

a. $T(n)=4T(\frac{n}{2})+n$ .. $T(1)=1$

$a=4$ $b=2$ $c=\log_b^a=\log_2^4=2$ $\Theta(n)^2>O(n) \Rightarrow T(n)=\Theta(n^2)$

b. $T(n)=4T(\frac{n}{2})+n^2$ $T(1)=1$

$a=4$ $b=2$ $c=\log_b^a=2$ $\Theta(n^2)=\Theta(n^L) \rightarrow T_n=\Theta(n^2\log n)$

c. $T(n)=4T(\frac{n}{2})+n^3$ $T(1)=1$

$a=4$ $b=2$ $c=\log_2^4=2$ $o(n^2)$, $o(n^3) \rightarrow T(n)=o(n^3)$

سوال ۹ . صفحه ۱۷۴ از همان الگوریتم mergesort کمک میگیریم با این تفاوت که هنگام merge کردن در هر آرایه یک اشاره گر داریم که به عنصر اول آرایه اشاره دارد و دو عنصر اول آرایه ها با هم مقایسه می شود که عنصری که کوچکتر است در آرایه اصلی جای گذاری شده و اشاره گر مربوط به آن یکی جلو می رود . در این الگوریتم ما هر بار عنصر آرایه ی دوم کوچکتر بود و می خواست در آرایه اصلی جایگزین شود به تعداد عناصر باقی مانده در آرایه ی اول باید به تعداد وارونه ها اضافه کرد . از آنجا تنها تفاوت با الگوریتم mergesort در یک جا است که آن هم پیچیدگی $\theta(1)$ دارد، پس پیچیدگی mergest یعنی $\theta(n\log n)$.

سوال ۸ . صفحه ۱۸۱ . تمرین ۲ فصل تقریبا شبیه الگوریتم partitioning عمل میکنیم . یک اشاره گر به ابتدای آرایه و یک اشاره گر به انتهای آن ایجاد میکنیم . هر بار اشاره گر اول را چک میکنیم اگر مقدار آن منفی بود که ایرادی ندارد . و اشاره گر اول را یکی جلو میبریم ولی اگر مقدار آن نامنفی بود ، عنصرهایی که دو اشاره گر ۱ و ۲ به آن اشاره میکنند را جابه جا میکنیم و اشاره گر دومی را یکی به عقب میبریم . تا زمانی این عمل را تکرار میکنیم که اشاره گر دوم از اشاره گر اول رد شود

الگوریتم Algorithm :

```
Array A[0,n-1]
i = 0 , j = n-1 ;   while (i<j) : { if (A[i] < 0)
i = i+1 ; else
swap (A[i], A[j]) ;   j = j-1 ;
```

Leaf counter (T)

سوال ۲ . صفحه ۱۸۵ . تمرین ۳ - ۵

complte recursively the number of leaves in a binary tree

Input: A binary tree T

output: The number of leaves in T   if T = Φ return 0 else return leafLounter (Tlefa) + Leafcamter (Tright)

غلط است. مثلا $\wedge$ باید در نظر بگیریم، به ۲ بخش و ۲ زیر درخت، تقسیم می‌کند برای هرکدام اجرا می‌کند. حالا این بخش‌ها ϕ نیستند هرکدام را بازهم به ۲ زیر درخت تقسیم می‌کنیم که این ۴ زیر درخت تهی هستند و پس 0 برگردانده می‌شود.

Leaf counter (T) if $T=\phi$

else f $T_{Left}=\phi$ and $T_{Right}=\phi$ return 1, *

else return leaf counter (T_{left}) + Leaf counter (T_{right});

*: یعنی برگ‌ها Note هایی هستند که زیر درخت چپ و راست تهی دارند. می‌توان نشان داد اگر درخت فقط اندازه ۱ باشد، مقصد یک‌بار برگ داده است.

if Number of Nodes (T) = 1

number of Nodes(T) return 1 + number of Nodes (T_{Right}) + number of Nodes (T_{Left});

سوال ۵ ↓

a. a b d e c f

b. d b e a c f

c. d e b f c a

و سوال ۷. صفحه ۱۸۵، تمرین ۳-۵ سوال ۷ ↓

سوال ۵. صفحه ۱۸۵. تمرین ۳-۵ الگوریتم میان ترتیب لیست را به صورت مرتب صعودی بر می‌گرداند برای اثبات این موضوع کافی است اثبات کنیم اگر k_1 از k_2 کوچک‌تر باشد، آنگاه در الگوریتم میان ترتیب گره k_1 زودتر از گره k_2 مشاهده می‌شود. برای این کار ابتدا گره k_3 را به صورت نزدیک‌ترین جد مشترک k_1 و k_2 در نظر می‌گیریم. اگر یکی از گره‌های k_1 یا k_2 جد دیگری باشد، گره k_3 همان گره جد می‌شود. اکنون ۲ حالت پیش می‌آید: ① در حالتی که گره k_3 متفاوت از گره k_2 است طبق تعریف درخت جستجو وجود دودویی گره k_1 در زیر درخت چپ k_3 و گره k_2 در زیر درخت راست k_3 قرار دارند. پس در الگوریتم میان ترتیب ابتدا گره k_1 و بعد گره k_2 مشاهده می‌شود. ② در حالتی که گره k_3 با گره k_2 یکی باشد، گره k_1 در زیر درخت چپ آن واقع شده و در الگوریتم میان ترتیب اول k_1 مشاهده می‌شود بعد k_2. در حالتی هم که k_3 با k_1 یکی باشد، گره k_2 صد در صد در زیر درخت راست k_1 وجود دارد و در الگوریتم میان ترتیب بعد از آن مشاهده می‌شود.

سوال ۲. صفحه ۱۹۱. تمرین ۴-۵

$a=a_1a_0$ $b=b_1b_0$ $c_2=a_1b_1$

$c_0=a_0b_0$

$c_1=(a_1+a_0)\times(b_1+b_0)-(c_2+c_0)$

$a = \underline{21}\ \underline{01}$ $\qquad$ $b = \underline{11}\ \underline{30}$

$c_2 = 21 \times 11$ $\qquad$ $c_0 = 01 \times 30$ $\qquad$ $c_1 = (21+01)\times(11+30) - (21\times 11 + 01\times 30)$

$c_1 = 22\times 41 - (21\times 11 + 01\times 30)$

$\rightarrow 21\times 11 \rightarrow \left.\begin{array}{l} c_2 = 2\times 1 = 2 \\ c_0 = 1\times 1 = 1 \\ c_1 = 3\times 2 - 3 = 3 \end{array}\right\} 21\times 11 = 231$

$\rightarrow 22\times 41 \rightarrow \left.\begin{array}{l} c_2 = 2\times 4 = 8 \\ c_0 = 2\times 1 = 2 \\ c_1 = 4\times 5 - 10 = 10 \end{array}\right\} 22\times 41 = 8\times 10^2 + 10\times 10 + 2 = 902$

$\rightarrow 2101 \times 1130 = 231 \times 10^4 + (902 - 231 - 30)\times 10^2 + 30 = 2374130$

سوال ۴. صفحه ۱۹۱. تمرین ۵-۴

$m_1 + m_4 - m_5 + m_7 = a_{00} b_{00} + a_{00} b_{11} + a_{11} b_{00} + \cancel{a_{11} b_{11}} + \cancel{a_{11} b_{10}} - \cancel{a_{11} b_{00}}$

$- a_{00} b_{11} - \cancel{a_{00} b_{11}} + a_{01} b_{10} + \cancel{a_{01} b_{11}} - a_{11} b_{10} - \cancel{a_{11} b_{11}} = a_{00} b_{00} + a_{01} b_{10}$

$m_3 + m_5 = a_{00} b_{01} - \cancel{a_{00} b_{11}} + \cancel{a_{00} b_{11}} + a_{01} b_{11} = a_{00} b_{01} + a_{01} b_{11}$

$m_2 + m_4 = a_{10} b_{00} + \cancel{a_{11} b_{00}} + a_{11} b_{10} - a_{11} b_{00} = a_{10} b_{00} + a_{11} b_{10}$

$m_1 + m_3 - m_2 + m_6 = \cancel{a_{00} b_{00}} + \cancel{a_{00} b_{11}} + \cancel{a_{11} b_{00}} + a_{11} b_{11} + \cancel{a_{00} b_{01}} - \cancel{a_{00} b_{11}}$

$- \cancel{a_{10} b_{00}} - \cancel{a_{11} b_{00}} + \cancel{a_{10} b_{00}} + a_{10} b_{01} - \cancel{a_{00} b_{00}} - \cancel{a_{00} b_{01}} = a_{10} b_{01} + a_{11} b_{11}$

$$\begin{bmatrix} \begin{matrix} 1 & 0 \\ 4 & 1 \end{matrix} & \begin{matrix} 2 & 1 \\ 1 & 0 \end{matrix} \\ \begin{matrix} 0 & 1 \\ 5 & 0 \end{matrix} & \begin{matrix} 3 & 0 \\ 2 & 1 \end{matrix} \end{bmatrix} * \begin{bmatrix} \begin{matrix} 0 & 1 \\ 2 & 1 \end{matrix} & \begin{matrix} 0 & 1 \\ 0 & 4 \end{matrix} \\ \begin{matrix} 2 & 0 \\ 1 & 3 \end{matrix} & \begin{matrix} 1 & 1 \\ 5 & 0 \end{matrix} \end{bmatrix}$$

سوال ۷. صفحه ۱۹۱. تمرین ۵-۴

$a_{00} = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix}$

$a_{10} = \begin{bmatrix} 0 & 1 \\ 5 & 0 \end{bmatrix}$

$$a_{01} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} \qquad b_{00} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} \qquad b_{10} = \begin{bmatrix} 2 & 0 \\ 1 & 3 \end{bmatrix}$$

$$a_{11} = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix} \qquad b_{01} = \begin{bmatrix} 0 & 1 \\ 0 & 4 \end{bmatrix} \qquad b_{11} = \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix}$$

$$m_1 = (a_{00} + a_{11}) * (b_{00} + b_{11}) = \begin{bmatrix} 4 & 0 \\ 6 & 2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 7 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 8 \\ 20 & 14 \end{bmatrix}$$

$$m_2 = (a_{10} + a_{11}) * b_{00} = \begin{bmatrix} 3 & 1 \\ 7 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 2 & 8 \end{bmatrix}$$

$$m_3 = a_{00}(b_{01} - b_{11}) = \begin{bmatrix} 1 & 0 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ -5 & 4 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -9 & 4 \end{bmatrix}$$

$$m_4 = a_{11}(b_{10} - b_{00}) = \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 6 & -3 \\ 3 & 0 \end{bmatrix}$$

$$m_5 = (a_{00} + a_{01}) b_{11} = \begin{bmatrix} 3 & 1 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 5 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 3 \\ 10 & 5 \end{bmatrix}$$

$$m_6 = (a_{10} - a_{00})(b_{00} + b_{01}) = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ -2 & -3 \end{bmatrix}$$

$$m_7 = (a_{01} - a_{11})(b_{10} + b_{11}) = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ -9 & -4 \end{bmatrix}$$

$$c_{00} = m_1 + m_4 - m_5 + m_7 = \begin{bmatrix} 4+6-8+3 & 8-3-3+2 \\ 20+3-10-9 & 14-6-4 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

$$c_{01} = m_3 + m_5 = \begin{bmatrix} -1+8 & 3 \\ -9+10 & 4+5 \end{bmatrix} = \begin{bmatrix} 7 & 3 \\ 1 & 9 \end{bmatrix}$$

$$c_{11} = m_1 + m_3 - m_2 + m_6 = \begin{bmatrix} 4-1-2+2 & 8-4+3 \\ 10-9-2-2 & 14+4-8-3 \end{bmatrix} = \begin{bmatrix} 3 & 7 \\ 7 & 7 \end{bmatrix}$$

سوال ۱ . صفحه ۱۹۷ . تمرین ۵ – ۵ مشابه الگوریتم Divide and Conq برای مسئله نزدیک نقاط عمل می‌کنیم . به این صورت که ابتدا به کمک یکی از الگوریتم های سریع مرتب سازی مانند Mergesort آرایه را به صورت صعودی مرتب می‌کنیم . بعد به صورت بازگشتی نزدیک ترین ۲ عدد در نیمه‌ی چپ آرایه و نزدیک ترین ۲ عدد در نیمه‌ی راست را پیدا می‌کنیم . تنها حالتی که یکی از این ۲ عدد جواب نیست حالتی است که آخرین عدد نیمه‌ی چپ و اولین عدد نیمه‌ی راست فاصله‌ی کمتری داشته باشند . پس min این ۳ عدد جواب ما خواهد بود . Algorithm .

closestNum(A[L...r]) if $L=r$ return -1 مسئله جواب ندارد

if $r-L=1$ return $A[r]-A[L]$

else return $\min\left\{ \begin{array}{l} \text{closestNum}(A[L..\lfloor \frac{L+r}{2} \rfloor]), \\ \text{closestNum}(A[\lfloor \frac{L+r}{2} \rfloor + 1 ...r]), \\ A[\lfloor \frac{L+r}{2} \rfloor + 1] \end{array} \right\}$

پیچیدگی : $T(n)=2T(n/2)+C \to a=2 \quad b=2 \quad d=0 \quad a>b^d$

$2>2^0$

تابع : $T(n)=\Theta(n^{\log_2 2})=\Theta(n)$ پیچیدگی کلی : $\underbrace{\Theta(n\log n)}_{\text{مربوط به مرتب سازی}}+\Theta(n)=\Theta(n\log n)$

سوال ۳ . صفحه ۱۹۷ . تمرین ۵ – ۵

$T(n)=2T(\frac{n}{2})+\underbrace{\frac{2n}{2}\log\frac{n}{2}}_{\text{۲ بار مرتب‌سازی ادغامی می‌نویسیم}}$

$T(2^k)=2T(2^{k-1})+2^k(k-1)=2[2T(2^{k-2})+2^{k-1}(k-2)]+2^k(k-1)=$

$2^2T(2^{k-2})+2^k(k-2)+2^k(k-1)=$

$2[2T(2^{k-3})+2^{k-2}(k-3)]+2^k(k-2)+2^k(k-1)=2^kT(2^{k-3})$

$+2^k(k-3)+2^k(k-2)+2^k(k-1)=$

$T(2^k)=2^iT(2^{k-i})+2^k(k-i)+2^k(k-i+1)+\dots+2^k(k-1)$

$$T(2^n) = 2^{k-1} T(2) + 2^k (1) + 2^k (2) + \dots + 2^n (k-1)$$

$$T(2^k) = 2^{k-1} \times 1 + 2^k (1 + 2 + \dots + k-1) = 2^{k-1} + 2^k \times \frac{(k-1)k}{2} =$$

$$T(2^k) = 2^{k-1} + 2^{k-1}(k^2 - k) = 2^{k-1}(k^2 - k + 1)$$

$$2^k = n \longrightarrow T(n) = \frac{n}{2}(\log^2 n - \log n + 1) \in \Theta(n \log^2_n)$$

سوال ۷. صفحه ۱۹۲ تمرین ۵-۵ تمام نقاط می‌توانند راس سوم مثلث با دو راس p_1 p_n باشند، بنابراین وقتی می‌خواهیم دورترین نقطه را انتخاب کنیم باید نقطه‌ای را انتخاب کنیم که مساحت مثلث p_1 p_{max} p_n ماکسیمم شود. مساحت مثلث با ۳ راس $p_1(x_1, y_1)$ و $p_2(x_2, y_2)$ و دیگر راس $p_{max}(x_3, y_3)$ از محاسبه ی دترمینان زیر بدست می‌آید.

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = x_2 y_3 - x_3 y_2 - x_1 y_3 + x_3 y_1 + x_1 y_2 - x_2 y_1$$

پس بنابراین $p_{max}(x_3, y_3)$ نقطه‌ای است که به ازای آن رابطه‌ی بالا بیشترین مقدار ممکن شود.